UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/808,470	03/25/2004	Tatsunori Kanai	251117US2SRD	6293

22850 7590 03/06/2009

OBLON, SPIVAK, MCCLELLAND MAIER & NEUSTADT, P.C.
1940 DUKE STREET
ALEXANDRIA, VA 22314

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2195	

NOTIFICATION DATE	DELIVERY MODE
03/06/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patentdocket@oblon.com
oblonpat@oblon.com
jgardner@oblon.com

Office Action Summary	Application No.	Applicant(s)	
	10/808,470	KANAI ET AL.	
	Examiner	Art Unit	
	KENNETH TANG	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 March 2004 and 29 September 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *8/9/04, 10/18/04, 12/2/04, 12/19/05, 8/21/06, 1/9/07, 6/2/08, 9/29/08*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-16 are presented for examination.

Specification

2. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 2 and 16 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 2 of U.S. Patent No. 7,418,705 B2.

4. It is noted that the preamble does not receive patentable weight. The table below has been constructed to illustrate how the limitations of claim 2 of the Instant Application are anticipated by claim 2 of U.S. Patent No. 7,418,705 B2 (bolded emphasis by Examiner to illustrate similarities):

INSTANT APPLICATION	**US 7,418,705 B2**
1. **A method** of assigning a plurality of threads to a plurality of processors, each of the threads being a unit of execution of a real-time operation, **the method comprising:**	1. **A method** of performing a real-time operation including a combination of a plurality of tasks, **the method comprising:** inputting structural description information and a plurality of programs describing procedures corresponding to the tasks, the structural description information indicating a relationship in input/output between the programs and including cost information concerning a time required for executing each of the programs, and coupling attribute information indicative of a coupling attribute between the programs; determining an execution start timing and execution term of each of a plurality of threads for execution of the programs based on

	the structural description information; **performing a scheduling operation of** assigning the threads to one or more processors of a plurality of processors according to a result of the determining;
selecting a tightly coupled thread group from among the threads based on coupling attribute information indicative of a coupling attribute between the threads, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other; and **performing a scheduling operation of dispatching the tightly coupled threads to several of the processors that are equal in number to the tightly coupled threads to simultaneously execute the tightly coupled threads by the several of the processors.**	**selecting a tightly coupled thread group from among the plurality of threads based on the coupling attribute information, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other;** **reserving execution terms of the tightly coupled threads in several processors of the plurality of the processors, the reserved execution terms having same execution start timing and the same term, the several processors being equal in number to the tightly coupled threads;** and **simultaneously executing the tightly coupled threads in reserved execution terms**

	by the several processors.
2. The method according to claim 1, wherein each of said plurality of processors includes a local memory, and the method further comprises mapping the local memory of one of the several of the processors, which executes one of the tightly coupled threads, in part of an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors.	**2. The method according to claim 1, wherein each of the several processors includes a local memory, and the method further comprises mapping the local memory of one of the several processors, which executes one of the tightly coupled threads, in part of an effective address space of another of the tightly coupled threads executed by another of the several processors.**

5. Claim 16 is rejected for the same reasons as stated in the rejection of claim 2. Although claim 16 of the Instant Application refers to a program which is stored in a computer-readable media and claim 2 of U.S. Patent No. 7,418,705 B2 refers to a method, it would have been obvious to one of ordinary skill in the art for the computer-readable media to contain the program that performs the method of claim 2 of U.S. Patent No. 7,418,705 B2 because it would provide a way for a computer to access the program to perform the method.

6. Claim 8 is rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 2 of U.S. Patent No. 7,418,705 B2.

7. It is noted that the preamble does not receive patentable weight. The table below has been constructed to illustrate how the limitations of claim 8 of the Instant Application are anticipated by claim 2 of U.S. Patent No. 7,418,705 B2 (bolded emphasis by Examiner to illustrate similarities)**:**

INSTANT APPLICATION	**US 7,418,705 B2**
7. **A method** of assigning a first thread and a second thread to a first processor having a local memory and a second processor having a local memory, the first thread and the second thread running in cooperation with each other, **the method comprising:**	1. **A method** of performing a real-time operation including a combination of a plurality of tasks, **the method comprising:** inputting structural description information and a plurality of programs describing procedures corresponding to the tasks, the structural description information indicating a relationship in input/output between the programs and including cost information concerning a time required for executing each of the programs, and coupling attribute information indicative of a coupling attribute between the programs;

performing a scheduling operation of dispatching the first thread and the second thread to the first processor and the second processor to simultaneously execute the first thread and the second thread by the first processor and the second processor, and	determining an execution start timing and execution term of each of a plurality of threads for execution of the programs based on the structural description information; **performing a scheduling operation of assigning the threads to one or more processors of a plurality of processors** according to a result of the determining; selecting a tightly coupled thread group from among the plurality of threads based on the coupling attribute information, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other; **reserving execution terms of the tightly coupled threads in several processors of the plurality of the processors,** the reserved execution terms having same execution start timing and the same term, the several processors being equal in number to the

	tightly coupled threads; and **simultaneously executing the tightly coupled threads in reserved execution terms by the several processors.**
mapping the local memory of the second processor, which executes the second thread, in an effective address space of the first thread executed by the first processor. **8. The method according to claim 7, further comprising mapping the local memory of the first processor, which executes the first thread, in an effective address space of the second thread executed by the second processor**	**2.** The method according to claim 1, wherein each of the several processors includes a local memory, and **the method further comprises mapping the local memory of one of the several processors, which executes one of the tightly coupled threads, in part of an effective address space of another of the tightly coupled threads executed by another of the several processors.**

8. Claim 10 is rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 4 of U.S. Patent No. 7,418,705 B2.

9. It is noted that the preamble does not receive patentable weight. The table below has been constructed to illustrate how the limitations of claim 10 of the Instant Application are anticipated by claim 4 of U.S. Patent No. 7,418,705 B2 (bolded emphasis by Examiner to illustrate similarities)**:**

INSTANT APPLICATION	**US 7,418,705 B2**
9. **A real-time processing system** that executes a plurality of threads, each of the threads being a unit of execution of a real-time operation, comprising: **a plurality of processors;**	3. An information **processing system** which performs a **real-time** operation including a combination of a plurality of tasks, the system comprising: **a plurality of processors;** means for storing structural description information and a plurality of programs describing procedures corresponding to the tasks, the structural description information indicating a relationship in input/output between the programs and including cost information concerning time required for executing each of the programs, and coupling

	attribute information indicative of a coupling attribute between the programs; means for determining an execution start timing and execution term of each of a plurality of threads for execution of the programs based on the structural description information; **means for performing a scheduling operation of** assigning the threads to at least one of the processors of a plurality of processors according to a result of the determining; **means for selecting a tightly coupled thread group from among the plurality of threads based on the coupling attribute information, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other, and** means for determining several
means for selecting a tightly coupled thread group from among the threads based on coupling attribute information indicative of a coupling attribute between the threads, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other; and **means for performing a scheduling**	

operation of dispatching the tightly coupled threads to several of the processors that are equal in number to the tightly coupled threads to simultaneously execute the tightly coupled threads by the several of the processors.	processors of the processors, to which the tightly coupled threads are to be assigned, to **simultaneously execute the tightly coupled threads by the several processors, the several processors being equal in number to the tightly coupled threads;** and means for simultaneously executing the tightly coupled threads in reserved execution terms by the several processors.
10. The real-time processing system according to claim 9, wherein each of said plurality of processors includes a local memory, and the system further comprises means for mapping the local memory of one of the several of the processors, which execute one of the tightly coupled threads, in part of an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors.	**4. The information processing system according to claim 3, wherein each of said plurality of processors includes a local memory, and the system further comprises means for mapping the local memory of one of the several processors, which executes one of the tightly coupled threads, in part of an effective address space of another of the tightly coupled threads executed by another of the several processors.**

10. Claim 13 is rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 8 of U.S. Patent No. 7,418,705 B2 in view of Day et al. (hereinafter Day) (US 2004/0249995 A1).

11. It is noted that the preamble does not receive patentable weight. The table below has been constructed to illustrate the limitations of claim 13 of the Instant Application which are contained in claim 8 of U.S. Patent No. 7,418,705 B2 (bolded emphasis by Examiner to illustrate similarities)**:**

INSTANT APPLICATION	**US 7,418,705 B2**
13. A **real-time processing system** that executes a **plurality of threads** including a first thread and a second thread which run in cooperation with each other, comprising: **a first processor** having a local memory; **a second processor** having a local memory; a shared **memory** shared by the first processor and the second processor;	7. An information **processing system** which performs a **real-time operation** including a combination of a **plurality of tasks**, the system comprising: **a plurality of processors;** **a storing unit** configured to store structural description information and a plurality of programs describing procedures corresponding to the tasks, the structural description information indicating a relationship in input/output between the

	programs and including cost information concerning time required for executing each of the programs, and coupling attribute information indicative of a coupling attribute between the programs; and
a scheduling unit configured to perform a scheduling operation of	**a scheduling unit configured to perform a scheduling operation of** assigning a plurality of threads for execution of the programs to at least one of the processors by determining an execution start timing and execution term of each of the threads based on the structural description information;
	a selector to select a tightly coupled thread group from among the plurality of threads based on the coupling attribute information, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other;
dispatching the first thread and the second thread to the first processor and the	a determining unit configured to **determine several processors of the processors, to which the tightly coupled**

second processor to simultaneously execute the first thread and the second thread by the first processor and the second processor; and	**threads are to be assigned, to simultaneously execute the tightly coupled threads by the several processors,** the several processors being equal in number to the tightly coupled threads; and the several processors configured to simultaneously execute the tightly coupled threads in reserved execution terms.
an address translation unit provided in the first processor, the address translation unit configured to convert an effective address space of the first thread executed by the first processor to a physical address space of the local memory of the second processor.	**8.** The information processing system according to claim 7, wherein each of said plurality of processors **includes a local memory**, and the system further comprises a **mapping unit configured to map the local memory of one of the several processors, which executes one of the tightly coupled threads, in part of an effective address space of another of the tightly coupled threads executed by another of the several processors.**

12. As shown above, US 7,418,705 B2 does not explicitly disclose both the first processor and the second processor having local memory and a shared memory for the first processor and second processor. However, Day teaches a first processor having a local memory (processor 1 having GMASK and GOFFSET control registers) (Fig. 1, col. 7, lines 22-26); a second processor having a local memory (processor 2 having GMASK and GOFFSET control registers) (Fig. 1, col. 7, lines 22-26); a shared memory shared by the first processor and the second processor (Global Registers 15, shared resource) (Fig. 1, col. 5, lines 12-13, col. 7, lines 22-24). One of ordinary skill in the art would have known to modify the computer system of US 7,418,705 B2 such that it would contain a memory structure that would include a local memory for each processor and a shared memory for the processors, as disclosed in Day. The suggestion/motivation for doing so would have been to provide the predicted result of improved memory management in the computer system (see lines 1-2 of the Abstract, page 1, [0007]).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1, 3-4, 6, 9, 11, and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Alfieri (US 5,745,778) in view of Strout, II et al. (hereinafter Strout) (US 5,339,415).

14. As to claim 1, Alfieri teaches a method of assigning a plurality of threads to a plurality of processors, each of the threads being a unit of execution of a real-time operation (see Abstract, Fig. 2, items 210-214, 220-223, 230-233, Fig. 1, items 100-103), the method comprising:

selecting a tightly coupled thread group from among the threads (related threads within a thread group) based on coupling attribute information indicative of a coupling attribute between the threads, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other (col. 2, lines 60-67, col. 3, lines 5-18, Fig. 4 items 401-402, col. 4, lines 23-29, col. 6, lines 31-58).

15. Alfieri is explicitly silent in performing a scheduling operation of dispatching the tightly coupled threads to several of the processors that are equal in number to the tightly coupled threads to simultaneously execute the tightly coupled threads by the several of the processors. However, Strout teaches a multiprocessor computer system that can execute simultaneously as multiple threads are each scheduled and dispatched, by the operating system, to a dedicated plurality of processors (see Fig. 1, Abstract, col. 1, lines 34-55, col. 5, line 37, col. 7, lines 10-15). Alfieri and Strout are analogous art because they are both in the same field of endeavor of thread/process scheduling on a multiprocessor and both attempting to solve the same problem of improving the processing performance within its multiprocessor system (see Abstracts of both

Alfieri and Strout). Therefore, one of ordinary skill in the art would have known to modify Alfieri's multiprocessing computer system, utilizing its tightly coupled threads, such that it performs a scheduling operation of dispatching the tightly coupled threads to several of the processors that are equal in number to the tightly coupled threads to simultaneously execute the tightly coupled threads by the several of the processors, as taught in the reference of Strout. The suggestion/motivation for doing so would have been to provide the predicted result of improving efficiency and processing performance by allowing for parallel execution (see Abstract, col. 1, lines 34-55, col. 5, lines 1-7). Therefore, it would have been obvious to one of ordinary skill in the art to combine Alfieri and Strout to obtain the invention of claim 1.

16. As to claim 3, Alfieri (col. 5, lines 53-62) and Strout (col. 5, lines 37-45, col. 1, lines 52-55) teaches wherein performing the scheduling operation includes reserving an execution term of each of the several of the processors.

17. As to claim 4, Strout teaches wherein each of the tightly coupled threads has context information indicating contents of a register (global registers 15 that may be shared by all processors or GMASK and GOFFSET control registers for each processor) (Fig. 1, item 15, col. 7, lines 22-32). Alfieri teaches wherein each of the tightly coupled threads has context information indicating contents of a local memory of one of the several of the processors (Fig. 1, item 108, 100).

18. As to claim 6, Strout teaches wherein the scheduling operation is performed by an operating system executed by one of said plurality of processors (col. 5, lines 37-45, col. 1, lines 52-55).

19. As to claim 9, it is rejected for the same reasons as stated in the rejection of claim 1.

20. As to claim 11, it is rejected for the same reasons as stated in the rejection of claim 3.

21. As to claim 15, it is rejected for the same reasons as stated in the rejection of claim 1. In addition, Alfieri's system contains a computer-readable media that stores the instructions to perform the method of claim 1 (col. 2, lines 53-54).

22. Claims 2, 10, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Alfieri (US 5,745,778) in view of Strout, II et al. (hereinafter Strout) (US 5,339,415), and further in view of Day et al. (hereinafter Day) (US 2004/0249995 A1).

23. As to claim 2, Alfieri in view of Strout teaches a computer having a memory system, wherein each of said plurality of processors includes a local memory and a group of tightly

coupled threads is executed (see Alfieri, col. 2, lines 43-54 and 60-67). Alfieri in view of Strout's computer system is explicitly silent in mapping the local memory of one of the several of the processors, which executes one of the tightly coupled threads, in part of an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors. However, Day discloses a multiprocessor system utilizing conventional/well-known virtual memory schemes with management for its memory such that the local memory of the processors is mapped to an effective address space of a different thread executed by a different processor (page 2, [0017], lines 3-8, and see claim 24). The effective address space is utilized by both its first processor and its second processor. One of ordinary skill in the art would have known to modify the memory component of Alfieri in view of Strout's multiprocessor system such that it would include the memory management features of Day's multiprocessor system, specifically, mapping the local memory of one of the several of the processors to an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors. The suggestion/motivation for doing so would have been to provide the predicted result of improving memory management in the multiprocessing system by allowing for cooperation and/or for providing the capability to support full virtual memory semantics, as taught in Day (page 1, last line of [0005], and [0008], and [0016]). Therefore, it would have been obvious to one of ordinary skill in the art to combine Alfieri, Strout, and Day to obtain the invention of claim 2.

24. As to claim 10, it is rejected for the same reasons as stated in the rejection of claim 2.

25. As to claim 16, it is rejected for the same reasons as stated in the rejection of claim 2.

26. Claims 5 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Alfieri (US 5,745,778) in view of Strout, II et al. (hereinafter Strout) (US 5,339,415), and further in view of Galpin (US 7,043,728 B1).

27. As to claim 5, Alfieri in view of Strout is silent wherein said plurality of processors are electrically connected to a shared memory, the method further comprises selecting a loosely coupled thread group from among the threads based on the coupling attribute information, the loosely coupled thread group including a set of loosely coupled threads communicating through a buffer on the shared memory, and performing the scheduling operation includes dispatching the loosely coupled threads to one or more of the processors in accordance with a relationship in input and output between the loosely coupled threads.

28. However, Galpin teaches a plurality of processors electrically connected to a shared memory and selects a loosely coupled thread group from among the threads, the loosely coupled thread group including a set of loosely coupled threads communicating through a buffer on the shared memory, and performing the scheduling operation (Scheduler 54, Fig. 1) includes dispatching the loosely coupled threads (Dispatcher 58, Fig. 1) to one or more of the processors

in accordance with a relationship in input and output between the loosely coupled threads (serial processing of loosely coupled threads wherein a first thread/process can take up execution of a third sequence, while the second thread/process can take up execution of a fourth sequence) (lines 1-14 of the Abstract, col. 7, lines 19-25). One of ordinary skilled in the art would have known to modify Alfieri in view of Strout's thread processing system such that it would have its coupling attribute information identify and selecting a loosely coupled thread group (in addition to it already doing so with tightly coupled thread group) using a plurality of processors electrically connected to a shared memory, and wherein the loosely coupled thread group includes a set of loosely coupled threads communicating through a buffer on the shared memory, and performing the scheduling operation, which includes dispatching the loosely coupled threads to one or more of the processors in accordance with a relationship in input and output between the loosely coupled threads, as taught in Galpin's thread processing system. The suggestion/motivation for doing so would have been to provide the predicted result of allowing the system to utilize independent threads with the capability to execute in a serial fashion, thus allowing the system to utilize the benefits of serial execution such as reducing complexity, cost, etc. Therefore, it would have been obvious to one of ordinary skill in the art to combine Alfieri, Strout, and Galpin to obtain the invention of claim 5.

29. As to claim 12, it is rejected for the same reasons as stated in the rejection of claim 5.

30. Claims 7-8 and 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Strout, II et al. (hereinafter Strout) (US 5,339,415) in view of Day et al. (hereinafter Day) (US 2004/0249995 A1).

31. As to claim 7, Strout teaches a method of assigning a first thread and a second thread to a first processor having a local memory and a second processor having a local memory, the first thread and the second thread running in cooperation with each other (see Abstract), the method comprising:

performing a scheduling operation of dispatching the first thread and the second thread to the first processor and the second processor to simultaneously execute the first thread and the second thread by the first processor and the second processor (see Fig. 1, Abstract, col. 1, lines 34-55, col. 5, line 37, col. 7, lines 10-15).

32. Strout is explicitly silent in mapping the local memory of the second processor, which executes the second thread, in an effective address space of the first thread executed by the first processor. However, Day discloses a multiprocessor system utilizing conventional/well-known virtual memory schemes with management for its memory such that the local memory of the processors is mapped to an effective address space of a different thread executed by a different processor (page 2, [0017], lines 3-8, and see claim 24). The effective address space is utilized by both its first processor and its second processor. One of ordinary skill in the art would have known to modify the memory component of Strout's multiprocessor system such that it would include the memory management features of Day's multiprocessor system, specifically, mapping

the local memory of the second processor, which executes the second thread, in an effective address space of the first thread executed by the first processor. The suggestion/motivation for doing so would have been to provide the predicted result of improving memory management in the multiprocessing system by allowing for cooperation and/or for providing the capability to support full virtual memory semantics, as taught in Day (page 1, last line of [0005], and [0008], and [0016]). Therefore, it would have been obvious to one of ordinary skill in the art to combine Strout and Day to obtain the invention of claim 7.

33. As to claim 8, Day teaches mapping the local memory of the first processor, which executes the first thread, in an effective address space of the second thread executed by the second processor (page 2, [0017], lines 3-8, and see claim 24). The effective address space is utilized and accessed by both its first processor and its second processor.

34. As to claim 13, Strout teaches a real-time processing system that executes a plurality of threads including a first thread and a second thread which run in cooperation with each other, comprising:

a first processor having a local memory (processor 1 having GMASK and GOFFSET control registers) (Fig. 1, col. 7, lines 22-26);

a second processor having a local memory (processor 2 having GMASK and GOFFSET control registers) (Fig. 1, col. 7, lines 22-26);

a shared memory shared by the first processor and the second processor (Global Registers 15, shared resource) (Fig. 1, col. 5, lines 12-13, col. 7, lines 22-24);

a scheduling unit configured to perform a scheduling operation of dispatching the first thread and the second thread to the first processor and the second processor to simultaneously execute the first thread and the second thread by the first processor and the second processor (see Fig. 1, Abstract, col. 1, lines 34-55, col. 5, line 37, col. 7, lines 10-15).

35. Strout is silent in explicitly teaching an address translation unit provided in the first processor, the address translation unit configured to convert an effective address space of the first thread executed by the first processor to a physical address space of the local memory of the second processor. However, Day discloses a multiprocessor system utilizing conventional/well-known virtual memory schemes with management for its memory such that the local memory of the processors is mapped to an effective address space of a different thread executed by a different processor (page 2, [0017], lines 3-8, and see claim 24). The effective address space is utilized by both its first processor and its second processor. Furthermore, a memory management unit (MMU) is used to perform the translation (page 1, [0016]). One of ordinary skill in the art would have known to modify the memory component of Strout's multiprocessor system such that it would include the memory management features of Day's multiprocessor system, specifically, an address translation unit provided in the first processor, wherein the address translation unit is configured to convert an effective address space of the first thread executed by the first processor to a physical address space of the local memory of the second processor. The suggestion/motivation for doing so would have been to provide the predicted result of improving memory management in the multiprocessing system by allowing for

cooperation and/or for providing the capability to support full virtual memory semantics, as taught in Day (page 1, last line of [0005], and [0008], and [0016]). Therefore, it would have been obvious to one of ordinary skill in the art to combine Strout and Day to obtain the invention of claim 13.

36. As to claim 14, Strout teaches the system further comprising another address translation unit provided in the second processor, the another address translation unit configured to convert an effective address space of the second thread executed by the second processor to a physical address space of the local memory of the first processor (page 2, [0017], lines 3-8, and see claim 24). The effective address space is utilized and accessed by both its first processor and its second processor.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

- **Cung et al. ("Strategies For the Parallel Implementation of Metaheuristics", May 3, 2001)** discloses two approaches for efficient parallel execution: (a) a group of "cooperative search threads" (the information collected along is disseminated and used by the other threads) within a task; (b) a group of "independent search threads" (the threads do not exchange any information they collect) within a task (see pages 8 and 10-14).

- **Bouraoui et al. (US 2002/0062432 A1)** discloses a multiprocessing system with memory management, executing a task consisting of a plurality of threads, such that the plurality of tasks/threads can utilize the same effective address ([0007], [0010], Fig. 5, Abstract).
- **Nozue et al. (US 5,890,189)** discloses memory management in a computer processing system realizing high speed execution and a proper and flexible memory access control for multiple programs sharing an identical logical address space (see first sentence of Abstract, col. 1, lines 12-39, col. 4, lines 23-58).
- **Alford et al. (US 6,904,595 B2)** discloses a multiprocessing system utilizing tasks, wherein the task is either a cooperative task comprising of cooperative threads or a preemptive task comprising of preemptive threads, based on a thread attribute table, a task status table, or a preempted thread table (see Abstract, Fig. 6, Fig. 13a-c).
- **D'Souza (US 6,052,707)** discloses the combination of preemptive scheduling with cooperative or non-preemptive scheduling. Tasks are divided into groups of interdependent tasks. The scheduler in the OS provides each group with a time slot of processor time and the tasks within the group are cooperatively scheduled (see Abstract).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KENNETH TANG whose telephone number is (571)272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Kenneth Tang/
Examiner, Art Unit 2195